Cadres argumentatifs avec nécessités

Farid Nouioua, Vincent Risch

Cadres argumentatifs avec nécessités

F. Nouioua V. Risch

LSIS - UMR CNRS 6168

Avenue Escadrille Normandie Niemen, 13397, Marseille Cedex 20, France

Résumé

[1] *Dans ce papier, nous introduisons les cadres argumentatifs avec nécessités (CANs), une extension des cadres argumentatifs de Dung (CAs) qui prend en compte une relation de nécessité comme sorte de support entre arguments (un argument est nécessaire pour un autre). Nous redéfinissons les sémantiques d'acceptabilité pour ces cadres étendus et nous montrons que la relation de nécessité assure une correspondance directe et immédiate entre un fragment des programmes logiques (PLs) et les CANs. Nous introduisons ensuite une généralisation des CANs qui étend la relation de nécessité pour lui permettre de porter sur des ensembles d'arguments. Nous présentons une adaptation naturelle des sémantiques d'acceptabilité à ce nouveau contexte et nous montrons que ce cadre généralisé permet de capter des PLs arbitraires.*

Mots Clef

argumentation abstraite, relation de nécessité, sémantiques d'acceptabilité, bipolarité, programmation logique.

Abstract

In this paper, we introduce argumentation frameworks with necessities (AFNs), an extension of Dung argumentation frameworks (AFs) taking into account a necessity relation as a kind of support between arguments (an argument is necessary for another). We redefine the acceptability semantics for these extended frameworks and we show how the necessity relation allows a direct and easy correspondance between a fragment of logic programs (LPs) and AFNs. We introduce then a further generalization of AFNs that extends the necessity relation to deal with sets of arguments.We give a natural adaptation of the acceptability semantics to this new context and show that the generalized frameworks allow to encode arbitrary LPs.

Keywords

abstract argumentation, necessity relation, acceptability semantics, bipolarity, logic programming.

1. Une version anglaise de ce papier a été acceptée à SUM'2011 (Dayton, USA, 10-12 octobre)

1 Introduction

Les cadres argumentatifs abstraits (CAs) initiés par Dung dans [10] ont récemment reçu un intérêt considérable. Malgré leur simplicité, les CAs fournissent un outil puissant qui capte différentes approches de raisonnement non-monotone. De nombreuses extensions ont été proposées aux CAs afin de les enrichir par de nouvelles fonctionnalités permettant de les utiliser dans de plus en plus de domaines clés de l'IA (prise de décision, gestion de l'incohérence, modélisation du dialogue ...). Comme l'activité d'argumenter implique souvent l'échange d'arguments pour ou contre une position donnée, l'un des concepts pertinents à formaliser dans un processus argumentatif est celui du support auquel nous nous intéressons dans cet article.

De façon générale, on peut distinguer deux principales approches pour le traitement de l'idée de support en argumentation abstraite. Dans la première approche, cette notion est prise dans le sens d'une inférence logique. Par exemple, [1] s'intéresse à la construction d'un CA à partir d'une base de connaissances logiques où suivant [2], les arguments sont de la forme $(support, conclusion)$, $support$ étant un ensemble cohérent minimal de formules de la base de connaissances qui infère la $conclusion$. La notion de support est ici un mécanisme interne à l'argument lui-même. Les CAs contraints proposés dans [9] ajoutent aux CAs classiques des contraintes propositionnelles qui manipulent les arguments comme des atomes. Le support opère alors à un niveau logique complémentaire au niveau abstrait.

La seconde approche, dans laquelle s'inscrit notre travail, consiste à ajouter aux CAs une nouvelle relation de support explicite entre arguments. Un travail bien connu dans ce domaine est celui sur les CAs bipolaires (CAB) [7] [8]. Son principal inconvénient est que les attaques supportées et indirectes, proposées dans ce modèle, peuvent conduire à des résultats contre-intuitifs si la relation de support a un sens précis comme celui de nécessité. A titre d'illustration, prenons le dialogue suivant entre deux agents A et B :

A : Je vais éclairer la salle ; **B :** la lampe ne s'allumera pas ; **A :** je vais ouvrir l'interrupteur et cela va allumer la lampe ; **B :** mais le compteur ne va pas fonctionner ; **A :** ah bon ? **B :** oui, l'électricien a détecté une panne dans le compteur.

Considérons les arguments suivants : la salle sera sombre (SE) ; la lampe s'allume (LA) ; l'interrupteur sera ouvert

(IO) ; le compteur fonctionne (CF), il y a une panne dans le compteur (P). Dans le CAB qui en résulte, on aura : P $attaque$ CF, LA $attaque$ SE, CF $supporte$ LA et IO $supporte$ LA. D'après [7], l'unique extension stable est $E = \{P, IO\}$ qui veut dire qu'il y a une panne dans le compteur et que l'interrupteur est ouvert. Nous nous attendons naturellement à ce que l'extension contienne également SE (la salle sera sombre), mais ce n'est pas le cas en raison de l'attaque supporté de SE par IO. Cependant, cette attaque ne s'explique que par le fait que IO supporte LA qui lui, attaque SE, mais puisque $LA \notin E$, il n'ya aucune raison de considérer l'attaque de SE par IO comme réussie. Ce problème persiste avec les différents types d'extensions présentées dans [7], mais peut-être résolu en remarquant que la relation de support utilisée est une relation de nécessité. Le lecteur pourra vérifier que l'unique extension stable dans l'approche décrite dans ce papier est $\{P, IO, SE\}$ qui correspond bien à nos attentes.

Parmi les principaux travaux ultérieurs dans cette approche, on peut également citer le travail de [18] sur l'argumentation basée sur les évidences et plus récemment, le travail de [3] sur les supports déductifs et défaisables ainsi que les cadres dialectiques abstraits [6] (voir la section 6 pour une comparaison de notre travail avec ces travaux).

La relation de support peut avoir différentes significations pouvant conduire à des traitements complètement différents. Notre objectif ici n'est pas de fournir une énumération exhaustive de ces significations possibles. Nous allons plutôt nous concentrer sur deux d'entre elles qui sont très intuitives : la nécessité et la suffisance. Dans les CAs, "a attaque b" est interprété par : "si a est accepté alors b n'est pas acceptée". De façon analogue, nous allons interpréter "a est nécessaire pour b " exactement comme "b est suffisant pour a" par : "si b est accepté alors a est accepté". Ainsi, sans perdre de généralité, nous pouvons nous concentrer sur une seule de ces deux relations. Nous avons choisi la relation de nécessité en raison de sa pertinence pour les PLs (voir les sections 4 et 5). La motivation de ce travail est de proposer de nouvelles méthodes directes et simples pour relier les systèmes d'argumentation et les PLs. Ce genre de traductions est très bénéfique : entre autres, il permet de réutiliser dans un formalisme les méthodes et les algorithmes développés dans l'autre formalisme et aussi de bénéficier des derniers développements importants dans le domaine de l'ASP pour la mise en œuvre efficace des systèmes argumentatifs. Pour ce faire, nous nous proposons dans cet article d'étendre les CAs de Dung avec une relation de nécessité et de considérer les points suivants : (1) nous mettons en évidence le type d'interactions qui résultent entre les nécessités et les attaques ; (2) à la lumière de ces interactions, nous redéfinissons les sémantiques d'acceptabilité dans le nouveau contexte. Comme les nécessités sont considérées au même niveau que les attaques, au lieu de proposer de nouvelles alternatives, nous optons pour l'intégration de la relation de nécessité aux définitions des principales sémantiques existantes. Mais cette intégration engendre ces propres problèmes auxquels nous apportons des solutions adéquates ; (3) Nous considérons ensuite la question de la liaison avec les PLs. Cette question initiée dans [10] continue à représenter un sujet de recherche actif et intéressant (voir par exemple [11], [16], [20]). Ici, nous montrons que l'ajout de la relation de nécessité, non seulement ne remet pas en cause les liens existants entre les CAs et les PLs mais de plus, permet de les simplifier.

D'abord, nous rappelons quelques notions de base sur l'ASP et les CAs de Dung. Ensuite, nous introduisons les CANs : nous généralisons leur sémantique d'acceptabilité et nous montrons comment les représenter comme des CAs classiques. Après cela, nous discutons la correspondance entre les CANs et un fragment des PLs. Ensuite, nous introduisons une nouvelle extension des CANs qui généralise la relation de nécessité pour qu'elle puisse porter sur des ensembles d'arguments. Après l'adaptation des notions clés et des sémantiques d'acceptabilité à ce nouveau contexte, nous montrons que ces CANs généralisés captent proprement des PLs arbitraires. Enfin, une dernière section comprendra une discussion des principaux travaux liés ainsi que quelques perspectives de travaux futurs.

2 Eléments de base

2.1 Les answer sets et les ι-answer sets

Un *PL normal* est un ensemble fini de règles de la forme :

$$a_0 \leftarrow a_1, ..., a_m, not\ a_{m+1}, ..., not\ a_n \quad (0 \leq m \leq n)$$

où chaque a_i $(0 \leq i \leq n)$ est un atome. Soit une règle r, on définit : $tete(r) = a_0$, $corps^+(r) = \{a_1, ..., a_m\}$ et $corps^-(r) = \{a_{m+1}, ..., a_n\}$. Plus généralement, pour tout PL Π on a : $tete(\Pi) = \{tete(r) | r \in \Pi\}$, $corps^+(\Pi) = \bigcup_{r \in \Pi} corps^+(r)$ et $corps^-(\Pi) = \bigcup_{r \in \Pi} corps^-(r)$. Un PL Π est *basique* si $corps^-(\Pi) = \emptyset$. Un ensemble d'atomes X est clos sous un PL basique Π si, $\forall r \in \Pi$, si $corps^+(r) \subseteq X$ alors $tete(r) \in X$. $Cn(\Pi)$ désigne le plus petit ensemble d'atomes clos sous Π.

La *réduction* d'un PL Π relativement à un ensemble d'atomes X est le PL basique $\Pi^X = \{tete(r) \leftarrow corps^+(r) | r \in \Pi, corps^-(r) \cap X = \emptyset\}$. X est un *answer set* de Π ssi $X = Cn(\Pi^X)$ [13]. Les *règles génératrices* et *applicables* de Π relativement à X sont données respectivement par : $G_\Pi(X) = \{r \in \Pi | corps^+(r) \subseteq X, corps^-(r) \cap X = \emptyset\}$ et $Ap_\Pi(X) = \{r \in \Pi | corps^+(r) \subseteq X, corps^-(r) \cap X = \emptyset, tete(r) \in X\}$. On a toujours : $Ap_\Pi(X) \subseteq G_\Pi(X)$.

Les *ι-Answer sets* [12] sont proposés comme la contrepartie des extensions justifiées de Lukaszewicz [14] dans le domaine de la logique des défauts [19] afin de surmonter le problème de la non-modularité dans la construction d'un answer set qui nécessite l'examen de toutes les règles simultanément. Son avantage est que, contrairement aux answer sets, il est possible de construire un ι-answer set de façon incrémentale et de valider localement une construction partielle (pour plus de détails, voir [14], [12]).

Soit $Cn^+(\Pi) = Cn(\Pi^\emptyset)$. Un ensemble X d'atomes est un ι-answer set de Π si $X = Cn^+(\Pi')$ où Π' est un sous-ensemble $\subseteq$-maximal de Π satisfaisant les conditions :
$(C1)$: $corps^+(\Pi') \subseteq Cn^+(\Pi')$
$(C2)$: $corps^-(\Pi') \cap Cn^+(\Pi') = \emptyset$.
Pour un PL Π et un ensemble d'atomes X, nous avons les résultats suivants : (1) *Construction incrémentale* : si X vérifie $X = Cn^+(Ap(X))$ alors il existe un ι-answer set X' tel que $X \subseteq X'$; (2) *Existence* : Π possède au moins un ι-answer set ; (3) *Les answer sets sont des ι-answer sets* : si X est un answer set de Π alors X est un ι-answer set de Π, et si X est un ι-answer set de Π alors X est un answer set de Π ssi $Ap_\Pi(X) = G_\Pi(X)$ [2].

2.2 Les cadres argumentatifs de Dung

Un CA de Dung [10] est défini par un couple $F = \langle A, R\rangle$ où A est un ensemble d'arguments et R est une relation binaire d'attaque sur A. Un ensemble $S \subseteq A$ attaque un argument b (noté $S\ R\ a$) ssi $\exists a \in S$ tel que $a\ R\ b$ [3]. S est *sans conflit* ssi il n'existe pas $a, b \in S$ avec $a\ R\ b$. Les sous-ensembles sans conflit maximaux de A sont appelés les *extensions naïves* [4] et ils représentent une première façon de construire des ensembles d'arguments acceptables collectivement. De nombreuses autres sémantiques d'acceptabilité ont été proposées dans [10]. Parmi elles, nous nous concentrerons dans cet article sur les sémantiques suivantes largement utilisées en pratique : (1) S est un *ensemble admissible* ssi S est sans conflit et $\forall a \in A\backslash S$, $\forall b \in S$, si $a\ R\ b$ alors $S\ R\ a$; (2) une *extension préférée* est un ensemble admissible maximal ; (3) S est une *extension stable* ssi S est sans conflit et $\forall a \in A\backslash S$, $S\ R\ a$.
Dung a montré dans [10] comment traduire tout CA en un PL et inversement. Si le passage d'un CA à un PL est immédiat, le passage dans l'autre sens est moins direct et nécessite un nombre d'arguments qui peut-être, en général, exponentiel par rapport au nombre d'atomes du PL [4].

3 Les CAs avec nécessités (CANs)

Considérons d'abord quelques définitions de base, en commençant par celle d'un CAN.

Définition 1. Un CAN est défini par $\langle A, R, N\rangle$ où A est un ensemble d'arguments, $R \subseteq A \times A$ (resp. $N \subseteq A \times A$) est une relation binaire d'*attaque* (resp. de *nécessité*) sur A. Pour $a, b \in A$, $a\ N\ b$ veut dire que b nécessite a.

De nouvelles attaques et nécessités émergent des attaques et nécessités directes. D'une part, la fermeture transitive de la relation de nécessité est interprétée comme une nécessité (si $a\ N\ b$ et $b\ N\ c$ alors a est nécessaire pour c) [5]. D'autre part, de nouvelles attaques émergent dans deux cas : si $a\ R\ c$ et $c\ N\ b$ alors a attaque b, et si $c\ R\ b$ et $c\ N\ a$ alors a attaque b (accepter a veut dire que c est également accepté, ce qui exclut b). En comparaison avec [7], notons que le premier cas correspond à l'attaque indirecte mais le deuxième cas ne correspond pas à l'attaque supportée.

Définition 2. Soit $\Delta = \langle A, R, N\rangle$ un CAN. Il y a une nécessité étendue de a à b ($a\ N^+\ b$) ssi il existe une séquence : $a_1\ N\ ...\ N\ a_n$ $(n \geq 2)$ où $a_1 = a$, $a_n = b$. Il y a une attaque étendue de b par a ($a\ R^+\ b$) ssi soit $a\ R\ b$, soit il existe $c \in A$ tel que : $a\ R\ c\ N^+\ b$ ou $c\ R\ b$ et $c\ N^+\ a$.

Il est clair qu'un CA classique est un cas particulier de CAN où $N = \emptyset$. Lorsque $N \neq \emptyset$, une exigence principale dans la définition de l'acceptabilité est d'éviter les cycles de nécessités qui reflètent une sorte d'interblocage vu comme une forme d'argument fallacieux (argument circulaire : "*begging the question*"). Le besoin d'exclure ces cycles est dû au sens particulier que nous voulons attribuer à "$a\ N\ b$" et qui veut dire : "pour obtenir b, il faut d'abord obtenir a". Une justification plus technique est que dans les PLs correspondant à des CANs (voir la section 4) de tels cycles sont effectivement exclus dans la définition des answers sets. Enfin, si le sens pris pour la relation de nécessité permet de tolérer les cycles, le traitement sera simplifié en enlevant juste les conditions d'évitement de ces cycles. De ce point de vue, notre approche est donc plus générale.

Définition 3. Soit $\Delta = \langle A, R, N\rangle$ un CAN et $a \in A$. a est dit *hors cycle de nécessité* (noté : Hors-Cycle-N) ssi il n'est pas le cas que $a\ N^+\ a$ ou que $b\ N^+\ a$ et $b\ N^+\ b$ pour $b \in A$. Un ensemble $S \subseteq A$ est dit Hors-Cycle-N ssi tous ses éléments sont Hors-Cycle-N.

Nous sommes à présent prêts à introduire les notions clés de *cohérence* et de *cohérence forte*. Cette dernière notion va justement généraliser la notion d'absence de conflits introduite dans les CA classiques de Dung.

Définition 4. Soit $\Delta = \langle A, R, N\rangle$ un CAN et $S \subseteq A$. S est dit *cohérent* ssi S est Hors-Cycle-N et clos sous N^{-1} (si $a \in S$ et $b\ N\ a$ alors $b \in S$). S est dit *fortement cohérent* ssi S est cohérent et sans conflit par rapport à R.

La cohérence d'un ensemble S exclut les cycles de nécessités et garantit que pour tout $a \in S$, S contient tous les arguments nécessaires à a. Notons aussi que pour $N = \emptyset$, la cohérence forte se réduit à l'absence de conflits.

3.1 Sémantiques d'acceptabilité des CANs

Les sémantiques d'acceptabilité pour les CANs suivent en gros les mêmes principes que celles pour les CAs en utilisant la cohérence forte au lieu de l'absence de conflits. Les extensions naïves et stables sont définies comme suit :

2. Dans le reste du papier, on suppose que tout PL Π utilisé, vérifie : $corps^+(\Pi) \subseteq tete(\Pi)$. En fait, toute règle r telle que $corps^+(r) \cap tete(\Pi) = \emptyset$ n'est jamais appliquée et il est facile de montrer qu'en la retirant de Π, on ne remet en cause ni ses ι-answer sets ni ses answer sets.

3. Pour $S, S' \subseteq A$ et $a \in A$, nous noterons par $a\ R\ S'$ (resp. $S\ R\ S'$) le fait que a (resp. un argument de S) attaque un argument de S'

4. Voir la méthode de traduction d'un PL en un CA dans [10].

5. Tout comme la relation d'attaque, on n'exige aucune propriété particulière sur la relation de nécessité. Cependant, la nature clairement transitive de la nécessité est captée au niveau de la fermeture transitive N^+.

Définition 5. Soit $\Delta = \langle A, R, N\rangle$ un CAN et $S \subseteq A$. (1) S est une extension naïve de Δ ssi S est fortement cohérent maximal ; (2) S est une extension stable de Δ ssi S est fortement cohérent et pour tout $a \in A \setminus S$ soit $S\ R\ a$ soit $\exists b \in A \setminus S$ tel que $b\ N\ a$.

La proposition suivante caractérise les arguments à l'intérieur et à l'extérieur d'une extension stable.

Proposition 1. Soit $\Delta = \langle A, R, N\rangle$ un CAN et $S \subseteq A$. S est une extension stable de Δ ssi : (1) S est Hors-Cycle-N et (2) ($a \in S$) ssi ($\forall b \in A$, si $b\ R\ a$ alors $b \notin S$ et si $b\ N\ a$ alors $b \in S$).

La partie ($\Rightarrow$) de la condition (2) exprime que si a est dans S, alors tout argument qui l'attaque n'est pas dans S et tout argument qui lui est nécessaire est dans S. La partie ($\Leftarrow$) veut dire que si a n'est pas dans S, alors soit il est attaqué par S, soit il existe un argument qui n'est pas dans S et qui est nécessaire pour a. La relation entre les extensions naïves et stables est exprimée par la proposition suivante.

Proposition 2. Chaque extension stable est naïve, mais l'inverse n'est pas vrai.

Revenons maintenant à la définition des ensembles admissibles et de la sémantique préférée pour les CANs en utilisant les notions de cohérence et de cohérence forte.

Définition 6. Soit $\Delta = \langle A, R, N\rangle$ un CAN et $S \subseteq A$. S est un ensemble admissible de Δ ssi S est fortement cohérent et si $b\ R\ S$ alors pour tout sous-ensemble cohérent $S' \subseteq A \setminus S$ tel que $b \in S'$ on a $S\ R\ S'$. Une extension préférée est un ensemble admissible maximal.

La définition 6 peut être vue comme une extension de la définition de Dung pour l'admissibilité où l'absence de conflit est remplacée par la cohérence forte et l'autodéfense concerne les attaques étendues. De plus, l'autodéfense n'est requise que contre les arguments qui sont Hors-Cycle-N. C'est ce qu'énonce la proposition suivante.

Proposition 3. Soit $\Delta = \langle A, R, N\rangle$ un CAN et $S \subseteq A$. S est un ensemble admissible de Δ ssi S est fortement cohérent et pour tout $b \in A \setminus S$, si b est Hors-Cycle-N et $b\ R\ S$ alors, $S\ R^+\ b$.

Il convient de noter que les propriétés des sémantiques stable et préférée pour les CA classiques sont préservées ici. C'est le cas notamment de l'existence d'extensions préférées et du fait que toute extension stable est préférée.

Proposition 4. Soit $\Delta = \langle A, R, N\rangle$ un CAN. (1) Δ a au moins une extension préférée ; (2) Toute extension stable de Δ est préférée, l'inverse n'est pas vrai.

Exemple 1. Les figures 1-(1) et 1-(2) montrent deux exemples de CANs où les arcs continus (resp. discontinus) représentent les attaques (resp. les nécessités). Le système de la figure 1-(1) a trois extensions naïves, $\{r_2\}$, $\{r_1, r_3\}$ et $\{r_4, r_5\}$. Deux d'entre elles sont stables, $\{r_2\}$ et $\{r_4, r_5\}$. Vérifions par exemple que $S = \{r_4, r_5\}$ est bien une extension stable. Seul r_4 a des attaqueurs (r_1 et r_2) qui ne sont pas dans S et seul r_5 nécessite un argument (r_4) qui est dans S. De plus, tout argument à l'extérieur de S est soit attaqué par S (le cas de r_1, r_2 et r_3) soit nécessite des arguments à l'extérieur de S (le cas de r_1). Les ensembles admissibles de ce système sont $\{r_2\}$ et $\{r_4, r_5\}$ qui sont aussi des extensions préférées et stables. Vérifions par exemple que $\{r_2\}$ est admissible mais que $\{r_1, r_3\}$ ne l'est pas. r_2 est attaqué par r_5 et tous les ensembles cohérents contenant r_5 doivent contenir r_4 qui est attaqué par r_2. Pour $\{r_1, r_3\}$ on a : r_3 est attaqué par r_2 mais $\{r_2\}$ qui est un ensemble cohérent contenant r_2 n'est pas attaqué par $\{r_1, r_3\}$.

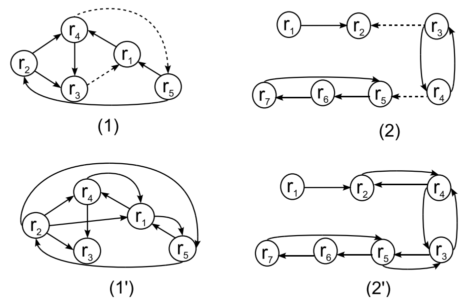

FIGURE 1 – Exemples de deux CANs et deux CAs

Le système de la figure 1-(2) possède huit extensions naïves dont une seule est stable : $\{r_1, r_3, r_6\}$, et deux extensions préférées : $\{r_1, r_3, r_6\}$ et $\{r_1, r_4\}$.

Notons que pour $N = \emptyset$, les extensions naïves, stables et préférées du CAN $\Delta = \langle A, R, \emptyset\rangle$ coïncident avec les extensions correspondantes du CA $F = \langle A, R\rangle$.

3.2 Les CANs vus comme des CAs

Dans cette section, nous montrons comment tout CAN est représenté par un CA ayant au plus le même nombre d'arguments. L'idée est de ne garder que les arguments qui sont Hors-Cycle-N et d'utiliser ensuite les attaques étendues comme relation d'attaque.

Définition 7. Soit $\Delta = \langle A, R, N\rangle$ un CAN. Δ est représenté par un CA $F_\Delta = \langle A_\Delta, R_\Delta\rangle$ tel que : $A_\Delta = \{a \in A|$ a est Hors-Cycle-N$\}$ et pour tout $x, y \in A_\Delta$, $x\ R_\Delta\ y$ ssi $x\ R^+\ y$ (R_Δ est la restriction de R^+ sur A_Δ).

Il se trouve que la notion de cohérence forte dans un CAN est plus contraignante que celle d'absence de conflit dans le CA correspondant. Cependant les extensions naïves, stables et préférées sont toutes conservés.

Proposition 5. Soit Δ un CAN, F_Δ son CA correspondant et $S \subseteq A$. On a : (1) si S est fortement cohérent dans Δ alors S est sans conflit dans F_Δ ; (2) S est une extension naïve (resp. stable, préférée) dans Δ ssi S est une extension naïve (resp. stable, préférée) dans F_Δ.

Exemple 1 (cont.). Soit Δ_1 et Δ_2 les CANs représentés dans les figures 1-(1) et 1-(2) respectivement. Les CAs correspondant F_{Δ_1} et F_{Δ_2} sont représentés dans les figures 1-(1') et 1-(2'). $\{r_2\}$, $\{r_1, r_3\}$, $\{r_4, r_5\}$ sont les extensions naïves et $\{r_4, r_5\}$, $\{r_2\}$ les extensions stables et préférées de Δ_1 et de F_{Δ_1}. On peut aussi vérifier que Δ_2 et F_{Δ_2} partagent les mêmes extensions naïves, stables et préférées.

Il est facile de vérifier que tout CAN se transforme en un temps polynomial en un CA. Cela veut dire d'un côté qu'il est possible de réutiliser les algorithmes développés dans le cadre des CAs pour les CANs et d'un autre côté que les CANs ne gagnent pas en expressivité par rapport aux CAs mais juste en concision (notons que les CANs généralisés (voir la section 5) gagnent en expressivité aussi).

4 Les CANs et les PLs

L'objectif de cette section est de tirer profit de la relation de nécessité pour proposer une nouvelle méthode de représentation d'un PL par un CAN. L'idée est de considérer une règle elle-même comme un argument. L'avantage de cette méthode est d'assurer une traduction immédiate assurant que le nombre d'arguments qui en résultent n'est plus exponentielle par rapport au nombre d'atomes, mais est égal au nombre de règles dans le PL. Cependant, en suivant cette méthode, les AFN ne captent qu'un fragment des PLs. Le fragment en question est noté $Frag$ et il contient les PLs dans lesquels chaque atome est donné par une seule règle du PL. Formellement, un PL Π est dans $Frag$ ssi $\forall r_1, r_2 \in \Pi$, si $r_1 \neq r_2$ alors $tete(r_1) \neq tete(r_2)$.
Soit $\Pi = \{r_1, ..., r_n\}$ un PL de $Frag$ contenant n règles. Le CAN correspondant est $\Delta_\Pi = \langle \Pi, R_\Pi, N_\Pi \rangle$ tel que : $\forall r_1, r_2 \in \Pi, r_1 \ R_\Pi \ r_2$ ssi $tete(r_1) \in corps^-(r_2)$ et $\forall r_1, r_2 \in \Pi, r_1 \ N_\Pi \ r_2$ ssi $tete(r_1) \in corps^+(r_2)$. Il existe alors une bijection entre les ι-answer sets (resp. answer sets) d'un PL de $Frag$ et les extensions naïves (resp. stables) dans le CAN correspondant.

Proposition 6. Soit Π un PL de $Frag$ et $\Delta_\Pi = \langle \Pi, R_\Pi, N_\Pi \rangle$ son CAN correspondant. Un ensemble X est un ι-answer set (resp. answer set) de Π, avec Π' le sous-ensemble maximal de Π correspondant ssi Π' est une extension naïve (resp. stable) de Δ_Π.

Exemple 2. Considérons le programme Π_1 :
$r_1 : a \leftarrow c, \, not\ e$
$r_2 : b \leftarrow not\ e$
$r_3 : c \leftarrow not\ d, \, not\ b$
$r_4 : d \leftarrow not\ a, \, not\ b$
$r_5 : e \leftarrow d$
Les ι-answer sets de Π sont $\{d, e\}$, $\{b\}$ et $\{a, c\}$. Pour $\{d, e\}$, le sous-ensemble maximal de Π_1 correspondant est $\Pi'_1 = \{r_4, r_5\}$. Puisque $corps^+(\Pi'_1) = \{d\} \subseteq \{d, e\}$ et $corps^-(\Pi'_1) \cap Cn^+(\Pi'_1) = \{a, b\} \cap \{d, e\} = \emptyset$, les conditions (C1) et (C2) sont vérifiées pour Π'_1 et on peut vérifier que Π'_1 est maximal. De plus, $G_{\Pi_1}(\{d, e\}) = Ap_{\Pi_1}(\{d, e\}) = \{r_4, r_5\}$, donc $\{d, e\}$ est aussi un answer set. De façon similaire, on peut vérifier que $\{b\}$ est un ι-answer set et un answer set et que $\{a, c\}$ est un ι-answer mais pas un answer set car $G_{\Pi_1}(\{a, c\}) \neq Ap_{\Pi_1}(\{a, c\})$. le CAN $\Delta_{\Pi_1} = \langle \Pi_1, R_{\Pi_1}, N_{\Pi_1} \rangle$ correspondant à Π_1 est celui représenté dans la figure 1-(1). Les trois ι-answer sets de Π_1, $\{d, e\}$, $\{b\}$ et $\{a, c\}$ ont, respectivement, $\{r_4, r_5\}$, $\{r_2\}$ et $\{r_1, r_3\}$ comme sous-ensembles maximaux correspondants de Π_1. Ces ensembles sont exactement les extensions naïves de Δ_{Π_1}. Parmi eux, seuls $\{r_4, r_5\}$ et $\{r_2\}$ sont des extensions stables de Δ_{Π_1} et ils correspondent bien aux deux answer sets de Π_1, $\{d, e\}$ et $\{b\}$.

Remarque 1. Inversement, tout CAN $\Delta = \langle A, R, N \rangle$ peut être traduit en un PL Π_Δ du fragment $Frag$. Chaque $r \in A$ donne naissance à un atome h_r et une règle r avec $tete(r) = h_r$, $corps^+(r) = \{h_s \mid s \ N \ r\}$, $corps^-(r) = \{h_s \mid s \ R \ r\}$. Pour montrer que cette traduction préserve la bijection entre les extensions naïves (resp. stables) de Δ et les ι-answer (resp. answer) sets de Π_Δ, il suffit de remarquer que la traduction de Π_Δ en utilisant la méthode décrite dans cette section donne justement Δ. Les résultats attendus sont alors obtenus par la proposition 6.

5 Les CANs généralisés

Afin de capter des programmes logiques arbitraires, nous étendons à présent les CANs de manière à ce que la relation de nécessité puisse exprimer le fait qu'un argument donné nécessite au moins un élément parmi un ensemble d'arguments. Les cadres qui en résultent sont appelés : les cadres argumentatifs avec nécessités généralisés (CANG).

Définition 8. Un CANG est défini par $\Delta = \langle A, R, DN \rangle$ où A est un ensemble d'arguments, R est une relation binaire d'attaque sur A et $DN \subseteq ((2^A \setminus \emptyset) \times A)$ est une relation de nécessité. $E \ DN \ b$ veut dire que b nécessite au moins un des arguments de E.

Nous adaptons légèrement la représentation graphique des CANs pour permettre à la relation de nécessité d'être définie sur des ensembles d'arguments (figure 2). La clôture sous DN^{-1} est adaptée au cas des CANGs comme suit :

Définition 9. Un ensemble $S \subseteq A$ est dit clos sous DN^{-1} ssi pour tout $a \in S$ et $E \subseteq A$ tel que $E \ DN \ a$, on a $E \cap S \neq \emptyset$.

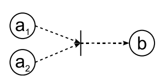

FIGURE 2 – Représentation de : $\{a_1, a_2\} \ DN \ b$

La définition 10 adapte la notion d'absence de cycles de nécessités au cas des CANGs. Les notions de cohérence simple et forte sont ensuite données par la définition 11.

Définition 10. Soit $\Delta = \langle A, R, DN \rangle$ un CANG, $S \subseteq A$ et $a \in S$. a est Hors-Cycle-N dans S ssi pour chaque $E \subseteq A$ tel que $E \ DN \ a$, soit $E \cap S = \emptyset$, soit $\exists b \in E \cap S$

tel que b est Hors-Cycle-N dans S. S est Hors-Cycle-N ssi tout $a \in S$ est Hors-Cycle-N dans S.

Définition 11. Soit $\Delta = \langle A, R, DN \rangle$ un CANG et $S \subseteq A$. S est cohérent ssi S est Hors-Cycle-N et clos sous DN^{-1}. S est fortement cohérent ssi il est cohérent et sans conflit par rapport à R.

Nous pouvons à ce stade généraliser les sémantiques d'acceptabilité de façon très analogue aux cas des CANs :

Définition 12. Soit $\Delta = \langle A, R, DN \rangle$ un $CANG$ et $S \subseteq A$. (1) S est une extension naïve de Δ ssi S est un sous-ensemble fortement cohérent maximal de A ; (2) S est une extension stable de Δ ssi S est fortement cohérent et pour tout $a \in A \setminus S$ soit $S\ R\ a$ soit $\exists E \subseteq A$ tel que $E\ DN\ a$ et $E \cap S = \emptyset$; (3) S est un ensemble admissible de Δ ssi S est fortement cohérent et si $b\ R\ S$ alors pour tout sous-ensemble cohérent $S' \subseteq A \setminus S$ tel que $b \in S'$ on a $S\ R\ S'$; (4) S est une extension préférée de Δ ssi S est un ensemble admissible maximal de Δ.

Les principaux résultats sur les sémantiques d'acceptabilité discutés plus haut restent valides pour les CANGs. Ces résultats sont résumés dans la proposition suivante :

Proposition 7. Soit $\Delta = \langle A, R, DN \rangle$ un $CANG$ et $S \subseteq A$. On a : (1) S est une extension stable de Δ ssi : (a) S est Hors-Cycle-N et (b) ($a \in S$) ssi ($\forall b \in A$, si $b\ R\ a$ alors $b \notin S$ et $\forall E \subseteq A$, si $E\ DN\ a$ alors $E \cap S \neq \emptyset$) ; (2) chaque extension stable de Δ est une extension naïve de Δ mais l'inverse n'est pas vrai ; (3) Il y a au moins une extension préférée pour Δ ; (4) Toute extension stable de Δ est préférée mais l'inverse n'est pas vrai.

Il n'est pas difficile de vérifier que les CANs sont un cas particulier des CANGs où les ensembles d'arguments nécessaires sont réduits à des arguments simples.

Exemple 3. Considérons les CANGs de la figure 3.

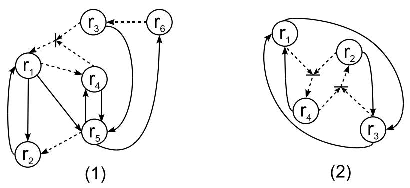

FIGURE 3 – Deux exemples de CANGs

Le système de la figure 3-(1) est un CANG en raison de la relation $\{r_3, r_4\}\ DN\ r_1$. $X_1 = \{r_1, r_4\}$ est clos sous DN^{-1} puisque le seul ensemble nécessaire pour r_4 (resp. r_1) est $\{r_1\}$ (resp. $\{r_3, r_4\}$) et $r_1 \in X_1$ (resp. $r_4 \in X_1$) mais X_1 n'est pas Hors-Cycle-N car ni r_1 ni r_4 n'est Hors-Cycle-N dans X_1. $X_2 = \{r_1, r_3\}$ est Hors-Cycle-N mais pas clos sous DN^{-1} : on a $\{r_6\}\ DN\ r_3$ mais $\{r_6\} \cap X_2 = \emptyset$. Finalement, $X_3 = \{r_1, r_3, r_6\}$ est à la fois Hors-Cycle-N et clos sous DN^{-1}. Le CANG de la figure 3-(1) possède deux extensions stables : $S_1 = \{r_1, r_3, r_4, r_6\}$ et $S_2 = \{r_2, r_5\}$. En effet, r_6 est Hors-Cycle-N (pas d'argument nécessaires pour r_6 dans S_1) ce qui rend r_3 Hors-Cycle-N, par conséquent r_1 est Hors-Cycle-N et enfin cela fait que r_4 est aussi Hors-Cycle-N. S_1 est donc Hors-Cycle-N. Tous les attaqueurs de S_1 sont à l'extérieur de S_1 et pour tout $a \in S_1$ et tout ensemble $E\ DN\ a$, on a $E \cap S \neq \emptyset$. D'un autre côté, r_2 et r_5 sont tous les deux attaqués par S_1. On peut vérifier que S_2 est une extension stable également. Notons que r_3 n'est pas accepté car : $\{r_6\}\ DN\ r_3$ et $\{r_6\} \cap S_2 = \emptyset$. On peut vérifier que les extensions préférées et stables de ce CANG coïncident.

Le CANG de la figure 3-(2) n'a aucune extension stable. En effet, on peut facilement voir que les seuls sous-ensembles fortement cohérents ici sont $\{r_1\}$, $\{r_3\}$ et $\emptyset$. Aucun de ces ensembles n'est stable et seul $\emptyset$ est admissible et représente la seule extension préférée. Prenons par exemple $\{r_1\}$. Il est attaqué par r_3 et r_4 mais $\{r_1, r_2, r_4\}$ est un sous-ensemble cohérent contenant r_4 sans être attaqué par $\{r_1\}$. $\{r_1\}$ n'est alors pas admissible et par conséquent n'est pas non plus ni préféré ni stable.

Montrons à présent comment les CANG permettent-ils de coder de façon simple tout PL. Soit Π un PL arbitraire contenant n règles, $\Pi = \{r_1, ..., r_n\}$. Le CANG qui correspond à Π est $\Delta_\Pi = \langle \Pi, R_\Pi, DN_\Pi \rangle$ tel que : ses arguments sont les règles de Π, sa relation d'attaque R_Π est définie par : $\forall r_1, r_2 \in \Pi, r_1\ R_\Pi\ r_2$ ssi $tete(r_1) \in corps^-(r_2)$ et sa relation de nécessité ND_Π est définie comme suit : soit E un ensemble de règles partageant la même tête que l'on dénote par $tete(E)$ et r une règle de Π. On considère que $E\ DN_\Pi\ r$ ssi $tete(E) \in corps^+(r)$. Nous avons alors une bijection entre les ι-answer sets (resp. answer sets) de Π et les extensions naïves (resp. stables) de Δ_Π.

Proposition 8. Soit Π un PL arbitraire et $\Delta_\Pi = \langle \Pi, R_\Pi, DN_\Pi \rangle$ le CANG correspondant. Un ensemble X est un ι-answer set (resp. answer set) de Π, avec Π' comme le sous-ensemble maximal de Π correspondant ssi Π' est une extension naïve (resp. stable) de Δ_Π.

Exemple 4. Considérons les deux PLs suivants :

$(\Pi_2)\ \ r_1 : a \leftarrow c,\ not\ b$	$(\Pi_3)\ \ r_1 : p \leftarrow not\ q$
$r_2 : b \leftarrow d,\ not\ a$	$r_2 : p \leftarrow q$
$r_3 : c \leftarrow e$	$r_3 : q \leftarrow not\ p$
$r_4 : c \leftarrow a,\ not\ d$	$r_4 : q \leftarrow p$
$r_5 : d \leftarrow not\ c$	
$r_6 : e \leftarrow not\ d$	

Le CANG $\Delta_{\Pi_2} = \langle \Pi_2, R_{\Pi_2}, DN_{\Pi_2} \rangle$ est représenté dans la figure 3-(1) et le CANG $\Delta_{\Pi_3} = \langle \Pi_3, R_{\Pi_3}, DN_{\Pi_3} \rangle$ dans la figure 3-(2). Le programme Π_2 possède deux ι-answer sets : $X_1 = \{a, c, e\}$ et $X_2 = \{b, d\}$ qui sont aussi ses seuls answer sets. Leurs sous-ensembles maximaux correspondants qui sont $S_1 = \{r_1, r_3, r_4, r_6\}$ et $S_2 = \{r_2, r_5\}$ respectivement, ont été prouvés dans l'exemple 3 être les extensions stables de Δ_{Π_2}. Le programme Π_3 a deux ι-

answer sets : $\{r_1\}$ et $\{r_3\}$ qui ne sont pas des answer sets [6].

Remarque 2. C'est la tête d'une règle qui détermine si elle attaque et/ou elle est nécessaire pour d'autres règles. Ainsi, pour un ensemble E de règles partageant la même tête, si une règle de E attaque une règle r alors toutes les règles de E doivent attaquer r, et si $E' \subseteq E$ est l'origine d'une relation de nécessité pour une règle r, alors tout l'ensemble E doit être l'origine d'une relation de nécessité pour r. C'est la raison pour laquelle pour pouvoir utiliser une méthode directe comme celle décrite dans la remarque 1 afin de traduire un CANG $\Delta = \langle A, R, DN \rangle$ en un PL Π_Δ, les conditions suivantes doivent être vérifiées : (1) si $E_1\ DN\ r_1$, $E_2\ DN\ r_2$ et $E_1 \neq E_2$ alors $E_1 \cap E_2 = \emptyset$; (2) si $E\ DN\ r$ pour $r \in A$ et $r'\ R\ s$ pour $r' \in E$ et $s \in A$ alors $r''\ R\ s$ pour tout $r'' \in E$.

Soit $\Delta = \langle A, R, DN \rangle$ un CANG satisfaisant les conditions précédentes, soit $E_1, ..., E_k$ tous les sous-ensembles de A tel que $E_i\ DN\ r$ pour un certain $r \in A$ et soit $P = \{r_{k+1}, ..., r_n\}$ les autres arguments éventuels de A qui ne sont impliqués dans aucun sous-ensemble E_i. Nous associons à chaque E_i un atome h_i ($1 \leq i \leq k$) et à chaque r_j (s'il en existe) de P un atome h_j ($k+1 \leq j \leq n$). Chaque argument $r \in A$ donne naissance alors à une règle r dans Π_Δ avec : $tete(r) = h_i$ ($1 \leq i \leq k$) si $r \in E_i$ pour un certain E_i et $tete(r) = h_j$ ($k+1 \leq j \leq n$) si $r = r_j$ pour un certain $r_j \in P$; $corps^+(r) = \{h_m \mid E_m\ DN\ r\}$ et $corps^-(r) = \{h_m \mid s\ R\ r\ et\ (s \in E_m\ ou\ s = r_m \in P)\}$. Pour montrer que cette traduction préserve une bijection entre les extensions naïves (resp. stables) de Δ et les ι-answer sets (resp. answer sets) de Π_Δ, il suffit de remarquer que la traduction de Π_Δ en utilisant la méthode décrite dans cette section, donne exactement Δ. Les résultats attendus sont alors obtenus par la proposition 8.

6 Discussion et travaux liés

Dans cet article, nous avons introduit les CANs qui étendent les CAs de Dung par une relation de support ayant le sens de nécessité. Nous avons redéfini les principales sémantiques d'acceptabilité et nous avons montré que les principales propriétés de celles-ci sont conservées dans ce nouveau contexte. Nous avons également proposé une nouvelle méthode qui tire profit de la relation de nécessité pour relier un fragment de PLs avec les CANs d'une manière très simple. Ensuite, afin de capter des PLs arbitraires, nous avons encore généralisé le cadre précédent pour permettre à la relation de nécessité de porter sur des ensembles d'arguments.

Nous avons déjà discuté dans la section 1. une principale critique des CABs présentés dans [7] [8]. Pour plus de différences techniques, notons que cette approche se limite à des systèmes acycliques alors que notre travail considère le cas général où les cycles de nécessités et d'attaques sont autorisés. De plus notre travail généralise la relation de support pour traiter des ensemble d'arguments et relie aussi bien des CANs que des CANGs avec les PLs.

Le travail présenté dans [3] démarre d'une critique des CABs sur le fait que leurs extensions ne garantissent pas l'admissibilité au sens de Dung. Afin de récupérer l'admissibilité, ils introduisent alors la notion de support déductif et d'attaque intermédiaire à la place de l'attaque indirecte. Il se trouve que le support déductif n'est rien d'autre que la relation de suffisance qui correspond à l'inverse de la relation de nécessité (voir la section 1). Ainsi, l'attaque intermédiaire et l'attaque supportée dans [3] correspondent respectivement au premier et au second cas de nos attaques étendues (voir définition 2). De plus, notre approche n'impose pas un seul type de support mais peut s'appliquer à un système où les deux types de supports (nécessité et suffisance) sont librement exprimés. Une étape préliminaire de pré-traitement peut alors ramener l'ensemble de toutes les relations à l'un des deux types. Notons aussi que contrairement à [3] où les extensions du nouveau carde bipolaire sont définies en passant par un méta modèle de Dung, notre approche consiste plutôt à étendre les définitions existantes afin d'intégrer la relation de nécessité sans être obligé de passer par un autre modèle pour définir les extensions.

Les cadres dialectiques abstraits (CDA) [6] représentent une généralisation puissante des CAs de Dung qui formalise l'idée de standards de preuve, largement étudiée dans le domaine du raisonnement légal. Cette idée est modélisée grâce à la notion de condition d'acceptation qui détermine si un argument est accepté en fonction du statut d'acceptation des arguments auxquels il est lié (ses parents). Les principaux résultats du papier concernent cependant une sous-classe des CDAs appelée les CDAs bipolaires (CDABs), où la relation entre un argument et un parent est toujours soit une attaque soit un support. Les CANs sont un cas particulier des CDAs, mais il n'en est pas ainsi pour les CANGs car un argument dans un CANG peut avoir comme parent un ensemble d'arguments tandis qu'un parent dans un CDA ne peut être qu'un argument unique. Cependant, il est possible de traduire un CANG en un CDAB en y ajoutant d'autres arguments. Une principale différence entre notre travail et [6] concerne la méthode utilisée pour généraliser les sémantiques d'acceptabilité. [6] adapte la réduction de Gelfond/Lifschitz utilisée dans le cadre des PLs afin d'éviter les cycles de nécessités. Dans notre travail, nous généralisons les définitions utilisées dans les CAs de Dung pour incorporer les nécessités et les traiter convenablement. Enfin, [6] montre comment représenter un PL à l'aide d'un CDA mais il ne considère pas la question inverse (la représentation des CDAs comme des PLs) alors que dans notre travail, nous précisons les contraintes que doit satisfaire un CANG afin d'assurer son encodage direct en un PL.

Une autre approche qui partage certaines caractéristiques

6. Malgré que : $Ap_{\Pi_3}(\{p,q\}) = G_{\Pi_3}(\{p,q\}) = \{r_2, r_4\}$, $\{p,q\}$ n'est pas un ι-answer set (et donc pas un answer set) car le sous-ensemble maximal correspondant $\Pi'_4 = \{r_2, r_4\}$ ne vérifie pas la condition (C1). Ceci est expliqué par le fait que dans Δ_{Π_3}, $\{r_2, r_4\}$ n'est pas fortement cohérent puisqu'il n'est pas Hors-Cycle-N.

avec la nôtre est celle de l'argumentation à base d'évidence [18]. Cette approche considère que seuls les arguments qui ont un certain support évident peuvent attaquer d'autres arguments. Un support évident est soit assuré directement par l'environnement à certains arguments particuliers soit à partir d'une chaîne de supports qui provient de tels arguments particuliers. Une idée très semblable est présente dans notre travail. En effet, pour assurer l'admissibilité d'un ensemble, nous devons garantir juste la réponse aux attaques venant des arguments qui sont Hors-Cycle-N, c'est à dire ceux qui n'ont pas besoin d'un support ou qui sont finalement supportés par des arguments qui n'ont pas besoin de support. Une perspective intéressante serait de voir comment pouvons-nous utiliser notre modèle dans le contexte du raisonnement à base d'évidence.

Notons que les CANs ne se réduisent pas à des CA contraints [9] où $a\ N\ b$ est remplacé par l'implication $b \Rightarrow a$. Prenons par exemple le cas de la sémantique stable. Le CAN $\langle A = \{a, b, c\}, R = \{(a, b)\}, N = \{(b, c)\}\rangle$ possède une seule extension stable $\{a\}$ mais le système contraint $\langle A, R, C = c \Rightarrow b\rangle$ n'a aucune extension stable : la seule extension stable de $\langle A, R\rangle$ est $\{a, c\}$ ne vérifie pas C).

De nombreux travaux se sont intéressés récemment au traitement de préférences entre arguments dans les CAs (voir par exemple [5] pour une brève synthèse). Le défi principal de ces travaux est de traiter les conflits potentiels entre les préférences et les attaques. Nous voulons étudier l'impact de l'ajout de préférences à des CAN(G) et d'exploiter les résultats obtenus dans le cadre des PLs avec des préférences sur les règles et/ou les atomes. Une autre perspective consiste à élaborer des formes adaptées de procédures de preuves dialectiques et/ou d'algorithmes d'étiquetage (voir [15]) pour les CAN(G)s. Le lien avec les PLs permettrait ensuite d'utiliser ces nouveaux algorithmes dans diverses applications. Par exemple, une procédure de preuve dialectique peut s'avérer plus adaptée que les solveurs ASP existants si nous voulons formuler des requêtes ciblées sur une base de connaissances codée par des PLs (pour une telle application, voir par exemple [17] qui propose un processus de raisonnement non-monotone pour détecter les causes d'accidents à partir de leurs descriptions textuelles). Enfin, nous envisageons également d'étendre notre cadre argumentatif pour capter le formalisme des PLs disjonctifs.

Références

[1] L. Amgoud, P. Besnard, Bridging the gap between abstract argumentation systems and logic, In : SUM'2009, Lecture Notes in Computer Science, Vol. 5785, pp. 12Ű27. Springer, 2009.

[2] P. Besnard, A. Hunter, *Elements of Argumentation*, The MIT Press, 2008.

[3] G. Boella, D.M. Gabbay, L. Van Der Torre, S. Villata, Support in Abstract Argumentation, In : COMMA'2010, pp. 40-51, IOS Press, 2010.

[4] A. Bondarenko, P.M. Dung, R. Kowalski, F. Toni, An abstract argumentation-theoretic approach to default reasoning, *Artificial Intelligence*, Vol. 93, pp. 63-101, 1997.

[5] J. Bourguet, L. Amgoud, R. Thomapoulos, Towards a unified model of preference-based argumentation, In : FoIKS'2010, pp. 326-344, 2010.

[6] G. Brewka, S. Woltran, Abstract Dialectical Frameworks, In : KR'2010, pp. 102-111, 2010.

[7] C. Cayrol, M.C. Lagasquie-Schiex, On the acceptability of arguments in bipolar argumentation frameworks. In : ECSQARU'2005, pp. 378-389, 2005.

[8] C. Cayrol and M.-C. Lagasquie-Schiex, Coalitions of arguments : A tool for handling bipolar argumentation frameworks, *Int. J. Intell. Syst.*, Vol 25(1), pp. 83-109, 2010.

[9] S. Coste-Marquis, C. Devred, P. Marquis, Constrained Argumentation Frameworks, In : KR'2006, pp. 112-122, AAAI Press, Lake District, 2006.

[10] P.M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence*, Vol 77(2), pp. 321-357, 1995.

[11] U. Egly, S.A. Gaggl, S. Woltran, Answer-set programming encodings for argumentation frameworks, *Argument and Computation*, Vol 1(2), 147-177, 2010.

[12] M. Gebser, M. Gharib, R. Mercer, T. Schaub, Monotonic Answer Set Programming, *Journal of Logic and Computation*, Vol 19(4), pp. 539-564, 2009.

[13] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, *New Generation Computing*, Vol 9, pp. 365-385, 1991.

[14] W. Łukaszewicz, Considerations on Default Logic : An Alternative Approach, *Computational Intelligence*, Vol 4, pp. 1-16, 1988.

[15] S. Modgil, M. Caminada, Proof theories and algorithms for abstract argumentation frameworks, In Rahwan, I. and Simari, G., eds., *Argumentation in Artificial Intelligence*, Springer, pp. 105-129, 2009.

[16] J.C. Nieves, U. Cortt'es, M. Osorio, Preferred extensions as stable models, *TPLP*, Vol 8(4), pp. 527-543, 2008.

[17] D. Kayser, F. Nouioua, From the description of an accident to its causes, *Artificial Intelligence*, Vol 173(12-13), pp. 1154-1193, 2010.

[18] N. Oren, T.J. Norman, Semantics for Evidence-Based Argumentation, In : COMMA 2008, Frontiers in Artificial Intelligence and Applications, pp. 276-284, 2008.

[19] R. Reiter, A Logic for Default reasoning, *Artificial Intelligence*, Vol 13(1-2), pp. 81-132, 1980.

[20] Wu, Y., Caminada, M., Gabbay, D. : Complete Extensions in Argumentation Coincide with 3-Valued Stable Models in Logic Programming, *Studia logica*, Vol 93(2-3), pp. 383-403, Springer, Heidelberg, 2009.